Interface Control Document

for

Using Third Parties to Deliver Infrastructure-to-Vehicle (I2V)

Version 1.0

Prepared by:

Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510

March 6, 2020

Prepared by ____ **Date** 03/06/2020
Cameron Mott
Senior Research Analyst

Approved by ____ **Date** 03/06/2020
Eric Thorn, Ph.D.
Manager R&D

Revision History

Version	Date	Author (s)	Comment
0.1	01/31/2020	SwRI	Initial Draft
1.0	03/06/2020	SwRI	Release

Table of Contents

List of Figures

List of Tables

1. Introduction

The participants in the Connected Vehicle (CV) Pool Fund Study (PFS) group are managing large amounts of data relevant to connected and non-connected vehicles. This data can be provided to vehicle operators through third-party companies or system integrators. To help scale and standardize the distribution of this data, the CV PFS team has funded a project called “Using Third Parties to Deliver Infrastructure-to-Vehicle (I2V).” This project is focused on the system engineering and standardization of data that is transmitted from States and other infrastructure owners and operators (IOOs) to third parties and Original Equipment Manufacturers (OEMs).

Stakeholder interviews have allowed the team to gather an understanding of the breadth of efforts that are already in place (or planned) regarding the usage of vehicular data by third-party companies and OEMs. As part of this project, the team has delivered a Survey of Current Status document detailing the maturity and future plans of CV data sharing activities. The team has also prepared a Concept of Operations (ConOps) document which describes the users, user needs, functions and features, operational scenarios, constraints, and context diagrams for the CV Data Framework (CVDF). This System Requirements Specification (SyRS) document has been generated based on the ConOps and ongoing collaboration with stakeholders. It includes the system requirements necessary to guide the development of a CVDF implementation to meet the needs of the stakeholders as described in the ConOps document. The final systems engineering document is this Interface Control Document (ICD) which defines the interfaces that are used to connect to the system.

1.1 Project Scope

The focus of this project is the standardization of communication from States and IOOs to third-party application providers or OEMs. Direct communication channels between States/IOOs and vehicles exist in the form of vehicle-to-everything (V2X) communication. This project recognizes that this will continue to occur, and in addition, there are auxiliary connections to vehicles and smart devices that are enabled through third-party connections. Existing communication from States/IOOs to third-party companies is based on an agency-by-agency and third party-by-third party arrangement, which is inefficient and complicated. This is represented in the “before” diagram shown in Figure 1.

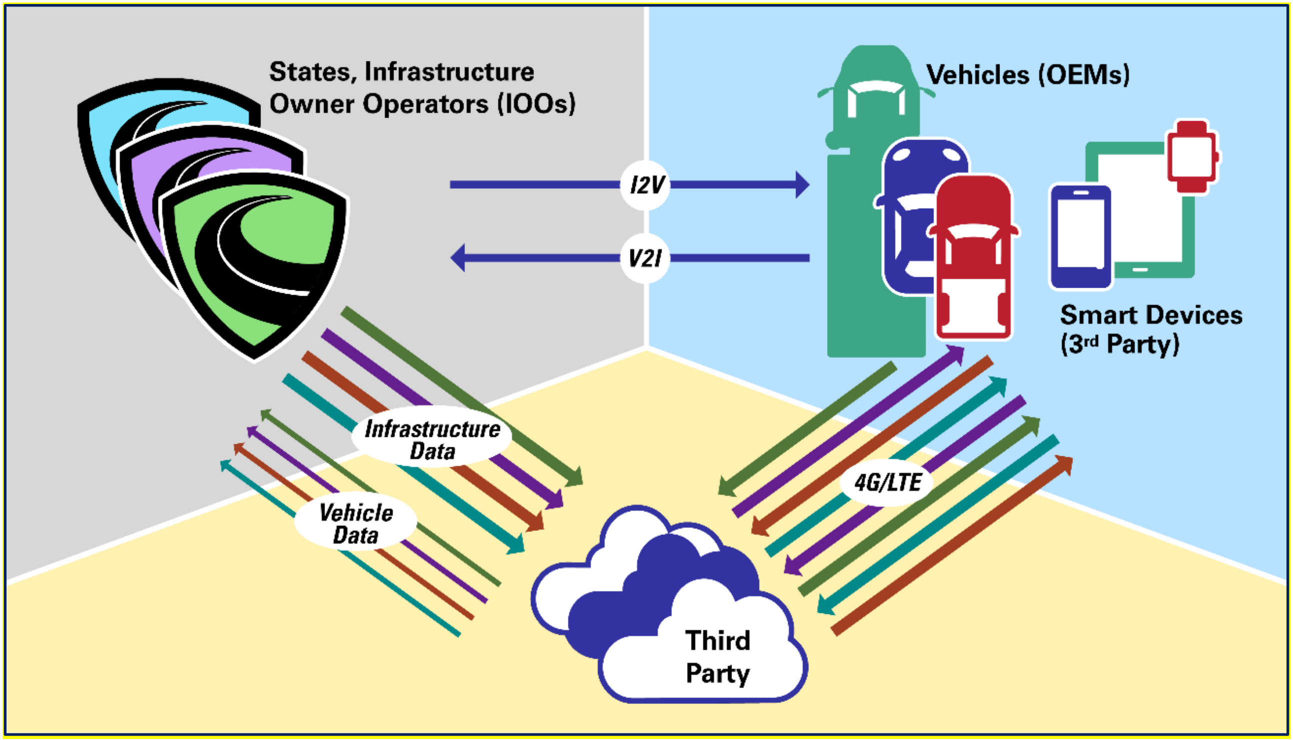

Figure 1: Example communication channels between states/IOOs, third-party companies, vehicles, and smart devices

This project will attempt to standardize the communication between States/IOOs and third-party companies. Standardizing this data will allow the multiple streams of data from IOOs and third parties to be consistent. The data will be provided via implementations that adhere to the CVDF defined through this project. Such implementations will result in the data exchanges shown in Figure 2. Through these efforts, the participants in the CV PFS and others can benefit from the usage of a standardized data distribution strategy.

The scope of the project includes the interactions between the CVDF and third parties or OEMs. The focus will be on the data that will be standardized and provided through the CVDF. Data (contained in payloads) provided through the CVDF is depicted in. Based on the Concept of Operations, valuable data for third parties includes (but is not limited to) Signal Phase and Timing (SPaT), MAP, Basic Safety Messages (BSMs), signal controller status, equipment location, and time synchronization data. Additional data that has been considered includes signal timing plans, pedestrian call and vehicle call information, presence detection, splits, signal cycles, work zone information, reduced speed limits, and queues. This document will propose the data types, data structures, and an example Application Programming Interface (API) that allows States/IOOs to communicate CV data to third parties and OEMs. Interactions between third parties and vehicles or user devices are considered out of scope for this project. Direct communication between States/IOOs and vehicles or user devices, as well as a back-end operation and administration of the data framework, are also considered out of scope.

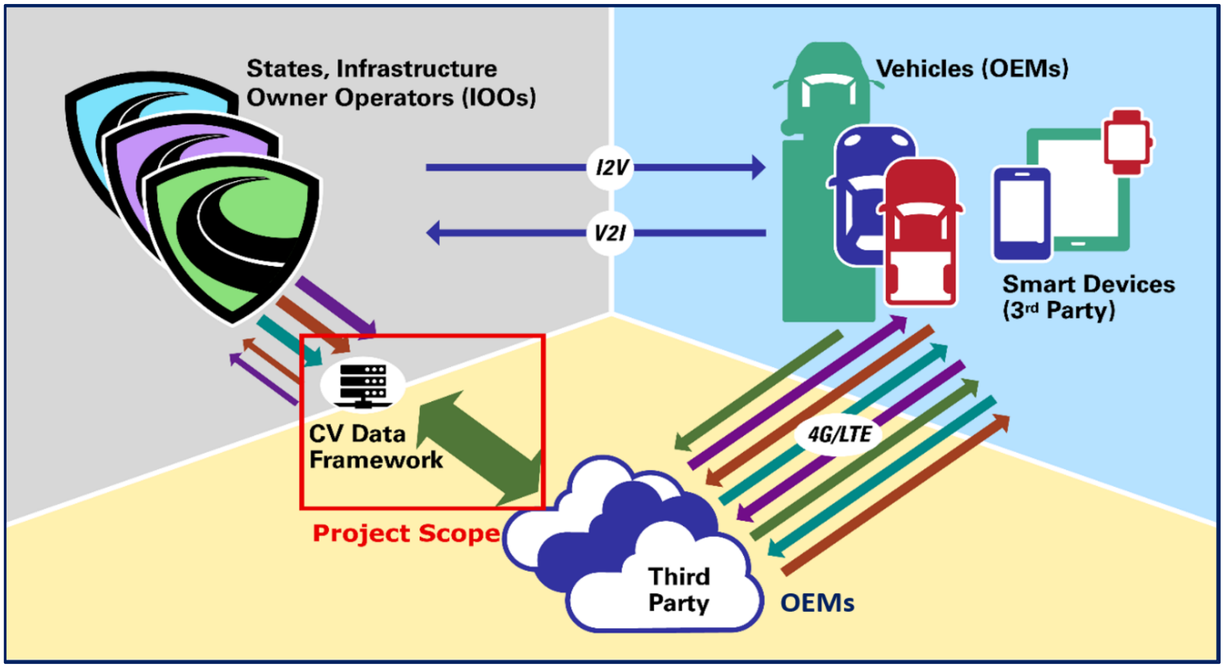

Figure 2: Project scope and interactions between the CVDFCV and other stakeholders

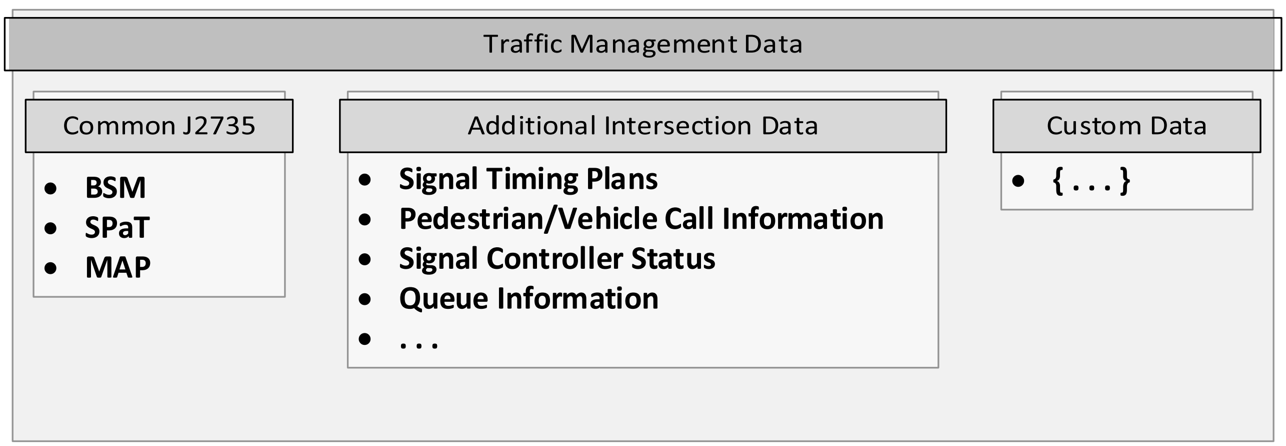

Figure 3: Example data that could be provided by implementations of the CVDF

1.2 Acronyms and Definitions

Table 1 provides a list of all the acronyms used in this document. Table 2 defines many of the terms that are used in this document where the authors identified some potential ambiguities or confusion.

Table 1: Acronym list

Acronym	Definition
API	Application Programming Interface
BSM	Basic Safety Message
ConOps	Concept of Operations
CV	Connected Vehicle
CVDF	CV Data Framework
HTTPS	Hypertext Transfer Protocol over SSL
I2V	Infrastructure-to-vehicle
ICD	Interface Control Document
ITS	Intelligent Transportation System
IOO	Infrastructure Owner and Operator
MAP	Not an acronym, refers to the SAE J2735 message that defines geographic information
NTCIP	National Transportation Communications for ITS Protocol
OAS	OpenAPI Specification
OEM	Original Equipment Manufacturer
OBU	On-Board Unit
PFS	Pooled Fund Study
REST	Representational State Transfer
SAE	Society of Automotive Engineering
SDK	Software Development Kit
SPaT	Signal Phase and Timing
SSL	Secure Sockets Layer
SyRS	System Requirements Specification
TLS	Transport Layer Security
TMC	Traffic Management Center
TMDD	Traffic Management Data Dictionary
TSC	Traffic Signal Controller

Table 2: Glossary of terms

Term	Definition
Function	A goal or objective accomplished by an implementation of a product or service.
Feature	A system or subsystem that is utilized in order to achieve a function.
Infrastructure Owner Operator	Regional authority that operates deployed field equipment and is usually responsible for enabling and managing traffic within a defined geographical area.
Requirement	A necessary condition that the system needs to meet to operate as designed.
Stakeholder	An organization with an interest in utilizing the framework proposed through this project.
System Integrator	An organization that maintains the hardware/software components that provide the functionality of the system.
Third-Party Application	A commercial product provided by a company working with at least one IOO to utilize CV data and provide information to a customer of their system. OEMs may be providing this commercial product as a feature that is integrated into their vehicles.
User	An operator of the system defined in this document or one that manages an external system.
Provider	An external system that has CV information and makes this available to the CVDF.
Consumer	An external system that is utilizing data from the CVDF.

1.3 Goals and Objectives

This document establishes the data types, data structures, and an example API that allows states and IOOs to communicate CV data to third parties and OEMs. The data types and format of the API is based on the understanding gained from stakeholder outreach and the creation of the ConOps and SyRS documents.

This document will inform the efforts towards standardization of the data types for communicating CV data from a CVDF. Through the use of this document, stakeholders such as third-party companies and OEMs can create a web client that can consume the desired data. States, IOOs, and contractors can use this document to create a web service that can publish CV information.

1.4 Document Overview

This document captures the data type for each of the data elements that are facilitated by an implementation of a CVDF. A message wrapper is defined that contains the metadata for messages that are handled by the CVDF. The specific data elements from SAE J2735, NTCIP 1202, and TMDD 3.03d are identified. Additionally, an example API is provided that addresses many of these data types as well as having the flexibility of accepting custom messages that are defined at a future time.

1.5 Related Documents

This document builds upon the ConOps and the SyRS documents that were created for this project.

1.5.1 Adjustments from Related Documents

The information from related documents are superseded by this document. One notable adjustment is listed below regarding a change to the handling of authentication. In previous documents, authentication was part of the message structure, but this is now handled outside of the message itself.

1.5.1.1 Authentication

Authentication is handled by the server's access manager and is done over Hypertext Transfer Protocol over SSL (HTTPS) with at least Transport Layer Security (TLS) v1.2. Prior to exchanging information with the CVDF, all participants must first register and be approved by the server administrators. This establishes trust between the participants and the server administrators. Administration of the server is outside of the scope of this document, but the high-level process is captured here for completeness. A participant (consumer/provider of CV data) registers with a username and password as well as a role. Participant roles may be simple, such as "provider/consumer" or more complex such as "consumer of SPaT, MAP, and NTCIP PhaseEntry data". Once approved, the participant is registered and will establish a connection with the CVDF over HTTPS. The endpoint provides a web token and establishes a symmetric key using an encryption algorithm such as SHA-256, AES-256, or other standardized algorithm. It is recommended that the web token also be assigned an expiration (N number of days) to further strengthen the security of the system. Once connected, the client can trigger Get/Post/Put/Delete commands via the API. When disconnecting, the client can actively disconnect or will be disconnected after an established period of inactivity.

2. Data Elements and Data Handling Strategies for External Interface

The list of data elements and messages described in this section are based on commonly deployed applications that utilize data to determine information such as traffic signal phase countdown and red light violation warning. The intent of the CVDF though is not to prescribe specific applications, but to provide a standardized means of sharing data that can then be used by consumers to support a range of applications, presumably many of which have not been developed or even envisioned yet. Based on feedback from stakeholders, the elements identified below are commonly understood from IOOs and typically available from current systems. As such, they provide a recommended minimal data set that is available for "Day 1" deployment and use by third-party consumers. Some of the data elements may not be required for certain applications (i.e., the NTCIP 1202 elements are not critical for displaying basic phase countdown for an intersection approach) but enable the same application(s) to run more reliably and/or intelligently or with greater confidence in what is presented to consumers. Similarly, there will likely be additional messages that can (and should) be standardized in the future when consumers identify the needs for the data based on specific use cases and applications. These new messages and/or elements can be added to new messages that are then incorporated into a CVDF implementation and potentially shared between implementors. The messages at the top level may not be required, however if implemented, they may be required to include specific elements within them.

The data elements in the external interface are wrapped in a high-level message with metadata that addresses the needs of the stakeholders as indicated in the ConOps. The messages include a header, a verification field, and a payload that contains other standardized messages or CVDF customized messages. Based on stakeholder engagement, valuable standardized messages include the SAE J2735 generic MessageFrame as well as the BasicSafetyMessage (BSM) MapData (MAP) and SignalPhaseAndTimingMessage (SPaT). NTCIP 1202 and TMDD v3.03d payloads and data elements are also identified as valuable CV data. Support for communicating these standardized messages is required for any CVDF implementation, but the CVDF does not need to verify the contents or structure of the standardized messages. Table 3 captures the high-level structure of a message sent by a CVDF to consumers. This table has been updated from the System Requirements, as the Authentication Token is handled as a part of the TLS v1.2 connection between a web client and the back-end access manager, and should not be coupled with the message.

Table 3: Example high-level message structure for an outgoing external message

Message Portion	Data Element
Header	Source
	Location/region
	Distribution restrictions
	Timestamp and time source
Payload	Standardized or Custom message
Verification	Signature

The high-level message structure is represented as a schema as indicated in Figure 4 below.


```
"header": {
  "locationOrRegion": "string",
  "distributionRestrictions": "string",
  "source": "string",
  "timeStamp": "2020-01-28T19:06:41.404Z",
  "timeSource": "string"
},
"payload": {
  "any": "string"
},
"verification": "string"
```


Figure 4: CVDF high-level message structure

A CVDF interface is required to support the transmission of standardized J2735 BSM, SPaT, MAP, and MessageFrame messages, standardized NTCIP 1202 messages, and standardized TMDD v3.03d messages, if and when the data are available. Aggregated messages with multiple payloads or a sub-set of messages are also supported.

Based on stakeholder engagement, the following message types are identified as valuable CV data and are required to be supported by an implementation of a CVDF.

- SAE J2735
  - SPaT
  - MAP
  - BSM
  - MessageFrame
- NTCIP 1202
  - Coordination Parameters
    - Split Table Entry
    - Pattern Table Entry
  - Overlap Parameters
    - Overlap Table Entry
    - Overlap Status Ground Table Entry
  - Phase Parameters
    - Phase Table Entry
    - Phase Status Group Table Entry
  - Ring Parameters
    - Ring Status Table Entry
  - Time Base Parameters
    - Time Base Pattern Sync Parameter
  - Channel Parameters

        - Channel Table Entry
    - Unit Parameters
        - Special Function Output Table Entry
- TMDD v3.03d
    - Intersection Signal
    - LCS
    - Link
    - Node

It is recommended that a CVDF interface also support custom messages that are identified as valuable based on stakeholder feedback:

- Geojson

When a payload is not identified as a directly-supported data type based on the API, the payload shall contain identifying information which specifies how the payload can be utilized by consumers. An enumeration that identifies the encoding and the message type is provided in pseudo code in Figure 5 below. This is indicated as an example and is expected to be agreed upon and expanded by both the providers and the consumers of any messages that are outside of the directly supported data types.

```
struct CVDFPayload {
    struct messageDescription description;
    string payloadBytes; //UTF-8
};

struct messageDescription {
    struct messageStandard fullStandard;
    enum e_encoding {Undefined=0, XER=1, BER=2, base64=3, HEX=4, ASCII=5, UPER=6}; //non inclusive
    enum e_type {Undefined=0, SPAT=1, MAP=2, BSM=3, TIM=4, MessageFrame=5, PSM=6, CUSTOM=7}; //non inclusive
};

struct messageStandard {
    enum e_standardBody {Undefined=0, SAE=1, IEEE=2, NIST=3, ISO=4, ITS=5} standardBody; //non inclusive
    enum e_standard {Undefined=0, NTCIP=1, TMDD=2, J2735=3} standard; //non inclusive
    enum e_standardVersion {Undefined=0, v1202=1, v303d=2, v201603=3} version; //non inclusive
    enum e_standardRev {Undefined=0, v01=1, v02=2, v03=3} revision; //non inclusive
};
```

Figure 5: CVDF Message low level formatting example

2.1 Data Normalization

The CV Data Framework recognizes the value of data normalization for end consumers, especially concerning the phase-centric nature of NTCIP 1202 compared to the intersection-centric approach for SAE J2735 messages. The intersection-centric information in SAE J2735 messages can be directly used by consumers, while the phase-centric information will need to be converted to information that correlates the data to an absolute location (such as Geographical Information

Systems [GIS]) before being used by consumers. This conversion is currently the responsibility of application logic hosted by third-party companies or OEMs. This conversion could be performed in a more efficient manner once it is supported by logic that is closer to the traffic signal controller. The normalization of data is expected to be dictated through efforts with standards bodies in the near future. When data has been normalized, the original provider and original format will need to be tracked. The CVDF supports the transmission of data both before and after this normalization has been established.

2.2 Data Streaming

To address the needs for real-time data streaming of elements such as SPaT and BSMs, a RESTful API with Get/Post/Put/Delete commands would not meet the needs for real-time data streaming. Instead, the expected mechanism would be to utilize web sockets to establish a connection from the providers to the CVDF and from the CVDF to any consumers. Alternatively, the CVDF could serve as a broker for direct connections, though this is discouraged as the connections would no longer be controlled through the CVDF. The web socket communication could utilize the same metadata and high-level message structure in order to retain the necessary information. The provider would initiate a web socket handshake to bridge from the RESTful (or GraphQL) HTTPS request to a secure web socket, establishing the necessary connection details. Once established, the CVDF can advertise the available real-time data and consumers can initiate a web socket handshake to receive the real-time streaming data. Communication between the client and the CVDF will follow typical web socket rules. Secure web socket connections are recommended, if a non-secure web socket is used it should be by agreement from both the provider and any consumers.

2.3 Applications

The valuable applications for stakeholders have been identified through previous project efforts and is represented in Table 4 below. Four of these applications (ECO, X-ECO, RLVW, PedX) are noted as high-value for CVDF implementations and the data elements to enable these applications are correlated in the sections below. For additional context on these applications, refer to the ConOps from this project.

Table 4: Software application reference table

Application Reference Name	Application Name	Description
ECO	Eco Approach and Departure at Signalized Intersections	An application that allows CVs to optimize the time spent and fuel consumed in order to traverse an intersection.
X-ECO	Extended Eco-Approach at Signalized Intersection	Similar to the ECOS but extends the scope of the application to encompass multiple intersections along a traveler's route, the intersection timing plans of those intersections, and historical traffic data related to the traveler's route.
RLVW	Red Light Violation	An application that will provide awareness to a connected vehicle and its driver that based on their current trajectory they will pass the stop bar of the intersection during a stop and remain phase.
PedX	Pedestrian in Crosswalk	An application that will provide awareness to a connected vehicle and its driver that the vehicle is approaching an area where a pedestrian has requested a walk signal.

2.4 SAE J2735 Data Elements

The data elements from SAE J2735 SPaT, MAP, and PSM are identified in Table 5, Table 6, and Table 7 respectively with traceability to the applications that utilize the data element. Each entry includes the name of the data element from the SAE J2735 standard and the data type, along with whether the SAE J2735 standard requires the data element (indicated in the Req column, with "Yes" indicating that the element is required by the J2735 standard, "No" indicating that it is optional). Based on the System Requirements and ConOps documents, SPaT, MAP, and BSM messages should be directly supported by CVDF implementations in order to enable the identified applications. The PSM is not transmitted by any known equipment, and is included below as an example for future deployments. BSM messages are expected to be passed along by the state/IOO with no modifications. Due to the lack of a need to modify the BSM, the data elements are not traced to applications and are not indicated below. Other J2735 messages and data elements can be supported but are not required.

Table 5: J2735 SPaT Data elements

SPaT Data Frames and Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
timeStamp=DE_MinuteOfTheYear	INT	No	X	X	X	X
name=DE_DescriptiveName	IA5String	No				
intersections=DF_IntersectionStateList	SEQUENCE	Yes	X	X	X	X
name=DE_DescriptiveName	IA5String	No				
id=DF_IntersectionReferenceID	SEQUENCE	Yes	X	X	X	X
region=DE_RoadRegulatorID	INT	No	X	X	X	X
id=DE_IntersectionID	INT	Yes	X	X	X	X
revision=DE_MsgCount	INT	Yes	X	X	X	X
status=DE_IntersectionStatusObject	BIT STRING	Yes	X	X	X	X
moy=DE_MinuteOfTheYear	INT	No	X	X	X	X
timeStamp=DE_Dsecond	INT	No	X	X	X	X
enabledLanes=DF_EnabledLaneList	SEQUENCE	No	X	X	X	X
LaneID=DE_LaneID	INT	Yes	X	X	X	X
states=DF_MovementList	SEQUENCE	Yes	X	X	X	X
movementName=DE_DescriptiveName	IA5String	No				
signalGroup=DE_SignalGroupID	INT	Yes	X	X	X	X
state-time-speed=DF_MovementEventList	SEQUENCE	Yes	X	X	X	X
eventState=DE_MovementPhaseState	ENUM	Yes	X	X	X	X
timing=DF_TimeChangeDetails	SEQUENCE	No	X	X	X	X
startTime=DE_TimeMark	INT	No				
minEndTime=DE_TimeMark	INT	Yes	X	X	X	X
maxEndTime=DE_TimeMark	INT	No			X	X
likelyTime=DE_TimeMark	INT	No				
confidence=DE_TimeIntervalConfidence	INT	No				

SPaT Data Frames and Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
nextTime=DE_TimeMark	INT	No			X	X
speeds=DF_AdvisorySpeedList	SEQUENCE	No				
type=DE_AdvisorySpeedType	ENUM	Yes				
speed=DE_SpeedAdvice	INT	No				
confidence=DE_SpeedConfidence	ENUM	No				
distance=DE_ZoneLength	INT	No				
class=DE_RestrictionClassID	INT	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
maneuverAssistList=DF_ManeuverAssistList	SEQUENCE	No		X	X	X
connectionID=DE_LaneConnectionID	INT	Yes		X	X	X
queueLength=DE_ZoneLength	INT	No			X	X
availableStorageLength=DE_ZoneLength	INT	No				
waitOnStop=DE_WaitOnStopline	BOOL	No				
pedBicycleDetect=DE_PedestrianBicycleDetect	BOOL	No		X		
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
maneuverAssistList=DF_ManeuverAssistList	SEQUENCE	No				
connectionID=DE_LaneConnectionID	INT	Yes				
queueLength=DE_ZoneLength	INT	No				
availableStorageLength=DE_ZoneLength	INT	No				
waitOnStop=DE_WaitOnStopline	BOOL	No				
pedBicycleDetect=DE_PedestrianBicycleDetect	BOOL	No		X		
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				

Table 6: J2735 MAP Data Elements and enabled applications

MAP Data Frames and Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
timeStamp=DE_MinuteOfTheYear	INT	No	X	X	X	X
msgIssueRevision=DE_MsgCount	INT	Yes	X	X	X	X
layerType=DE_LayerType	ENUM	No				
layerID=DE_LayerID	INT	No				
intersections=DF_IntersectionGeometryList	SEQUENCE	No	X	X	X	X
name=DE_DescriptiveName	IA5String	No				
id=DF_IntersectionReferenceID	SEQUENCE	Yes	X	X	X	X
region=DE_RoadRegulatorID	INT	No	X	X	X	X
id=DE_IntersectionID	INT	Yes	X	X	X	X
revision=DE_MsgCount	INT	Yes	X	X	X	X
refPoint=DF_Position3D	SEQUENCE	Yes	X	X	X	X
lat=DE_Latitude	INT	Yes	X	X	X	X
long=DE_Longitude	INT	Yes	X	X	X	X
elevation=DE_Elevation	INT	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
laneWidth=DE_LaneWidth	INT	No	X	X	X	X
speedLimits=DF_SpeedLimitList	SEQUENCE	No			X	X
type=DE_SpeedLimitType	ENUM	Yes				
speed=DE_Velocity	INT	Yes			X	X
laneSet=DF_LaneList	SEQUENCE	Yes	X	X	X	X
laneID=DE_LaneID	INT	Yes	X	X	X	X
name=DE_DescriptiveName	IA5String	No				
ingressApproach=DE_ApproachID	INT	No				
egressApproach=DE_ApproachID	INT	No				

MAP Data Frames and Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
laneAttributes=DF_LaneAttributes	SEQUENCE	Yes	X	X	X	X
directionalUse=DE_LaneDirection	BIT STRING	Yes	X	X	X	X
sharedWith=DE_LaneSharing	BIT STRING	Yes				
laneType=DF_LaneTypeAttributes	CHOICE	Yes	X	X	X	X
vehicle=DE_LaneAttributes-Vehicle	BIT STRING	Yes				
crosswalk=DE_LaneAttributes-Crosswalk	BIT STRING	Yes				
bikeLane=DE_LaneAttributes-Bike	BIT STRING	Yes				
sidewalk=DE_LaneAttributes-Sidewalk	BIT STRING	Yes				
median=DE_LaneAttributes-Barrier	BIT STRING	Yes				
striping=DE_LaneAttributes-Striping	BIT STRING	Yes				
trackedVehicle=DE_LaneAttributes-TrackedVehicle	BIT STRING	Yes				
parking=DE_LaneAttributes-Parking	BIT STRING	Yes				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
maneuvers=DE_AllowedManeuvers	BIT STRING	No	X	X	X	X
nodeList=DF_NodeListXY	CHOICE	Yes	X	X	X	X
nodes=DF_NodeXY	SEQUENCE	Yes				
delta=DF_NodeOffsetPointXY	SEQUENCE	Yes	X	X	X	X

MAP Data Frames and Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
node-XY1=DF_Node-XY-20b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B10	INT	Yes	X	X	X	X
y=DE_Offset_B10	INT	Yes	X	X	X	X
node-XY2=DF_Node-XY-22b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B11	INT	Yes	X	X	X	X
y=DE_Offset_B11	INT	Yes	X	X	X	X
node-XY3=DF_Node-XY-24b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B12	INT	Yes	X	X	X	X
y=DE_Offset_B12	INT	Yes	X	X	X	X
node-XY4=DF_Node-XY-26b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B13	INT	Yes	X	X	X	X
y=DE_Offset_B13	INT	Yes	X	X	X	X
node-XY5=DF_Node-XY-28b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B14	INT	Yes	X	X	X	X
y=DE_Offset_B14	INT	Yes	X	X	X	X
node-XY6=DF_Node-XY-32b	SEQUENCE	Yes	X	X	X	X
x=DE_Offset_B16	INT	Yes	X	X	X	X
y=DE_Offset_B16	INT	Yes	X	X	X	X
node-LatLon=DF_Node-LLmD-64b	SEQUENCE	Yes	X	X	X	X
lon=DE_Longitude	INT	Yes	X	X	X	X
lat=DE_Latitude	INT	Yes	X	X	X	X
attributes=DF_NodeAttributeSetXY	SEQUENCE	No				
localNode=DF_NodeAttributeXYList	SEQUENCE	No				
NodeAttributeXY=DE_NodeAttributeXY	ENUM	Yes				
disabled=DF_SegmentAttributeXYList	SEQUENCE	No				
NodeAttributeXY=DE_NodeAttributeXY	ENUM	Yes				

MAP Data Frames and Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
enabled=DF_SegmentAttributeXYList	SEQUENCE	No				
NodeAttributeXY=DE_NodeAttributeXY	ENUM	Yes				
data=DF_LaneDataAttributeList	SEQUENCE	No				
LaneDataAttribute=DF_LaneDataAttribute	CHOICE	Yes				
dWidth=DE_Offset_B10	INT	No				
dElevation=DE_Offset_B10	INT	No				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
computed=DF_ComputedLane	SEQUENCE	Yes	X	X	X	X
referenceLaneId=DE_LaneID	INT	Yes	X	X	X	X
offsetXaxis	CHOICE	Yes	X	X	X	X
small=DE_DrivenLineOffsetSmall	INT	Yes				
large=DE_DrivenLineOffsetLarge	INT	Yes				
offsetYaxis	CHOICE	Yes	X	X	X	X
small=DE_DrivenLineOffsetSmall	INT	Yes				
large=DE_DrivenLineOffsetLarge	INT	Yes				
rotateXY=DE_Angle	INT	No	X	X	X	X
scaleXaxis=DE_Scale_B12	INT	No	X	X	X	X
scaleYaxis=DE_Scale_B12	INT	No	X	X	X	X
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
connectsTo=DF_ConnectsToList	SEQUENCE	No		X	X	X
connectingLane=DF_ConnectingLane	SEQUENCE	Yes		X	X	X
lane=DE_LaneID	INT	Yes				
maneuver=DE_AllowedManeuvers	BIT STRING	Yes				
remoteIntersection=DF_IntersectionReferenceID	SEQUENCE	No				
region=DE_RoadRegulatorID	INT	No				

MAP Data Frames and Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
id=DE_IntersectionID	INT	Yes				
signalGroup=DE_SignalGroupID	INT	No	X	X	X	X
userClass=DE_RestrictionClassID	INT	No				
connectionID=DE_LaneConnectionID	INT	No		X	X	X
overlays=DF_OverlayLaneList	SEQUENCE	No				
laneID=DE_LaneId	INT	Yes				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
preemptPriorityData=DF_PreemptPriorityList	SEQUENCE	No				
zone=RegionalExtension	SEQUENCE	Yes				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				
roadSegments=DF_RoadSegmentList	SEQUENCE	No				
dataParameters=DF_DataParameters	SEQUENCE	No				
processMethod	IA5String	No				
processAgency	IA5String	No				
lastCheckedDate	IA5String	No				
geoidUsed	IA5String	No				
restrictionList=DF_RestrictionClassList	SEQUENCE	No	X		X	X
DE_RestrictionClassID	INT	Yes	X		X	X
DF_RestrictionUserTypeList	SEQUENCE	Yes	X		X	X
users=DF_RestrictionClassAssignment	SEQUENCE	Yes	X		X	X
id=DE_RestrictionClassID	INT	Yes	X		X	X
users=DF_RestrictionUserType	SEQUENCE	Yes	X		X	X
basicType=DE_RestrictionAppliesTo	ENUM	Yes	X		X	X
regional=1 to 4 x RegionalExtension	SEQUENCE	Yes				
regional=1 to 4 x RegionalExtension	SEQUENCE	No				

Table 7: Personal Safety Message Data Elements and enabled applications

PSM Data Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
basicType=DE_PersonalDeviceUserType	ENUM	Yes		X		
secMark=DE_DSecond	INT	Yes		X		
msgCnt=DE_MsgCount	INT	Yes		X		
id=DE_TemporaryID	OCTET STRING	Yes		X		
position=DF_Position3D	SEQUENCE	Yes		X		
lat=DE_Latitude	INT	Yes		X		
long=DE_Longitude	INT	Yes		X		
elevation=DE_Elevation	INT	No		X		
regional=RegionalExtension	SEQUENCE	No				
accuracy=DF_PositionalAccuracy	SEQUENCE	Yes		X		
semiMajor=DE_SemiMajorAxisAccuracy	INT	Yes		X		
semiMinor=DE_SemiMinorAxisAccuracy	INT	Yes		X		
orientation=DE_SemiMajorAxisOrientation	INT	Yes		X		
speed=DE_Velocity	INT	Yes		X		
heading=DE_Heading	INT	Yes		X		
accelSet=DF_AccelerationSet4Way	SEQUENCE	No				
long=DE_Acceleration	INT	Yes				
lat==DE_Acceleration	INT	Yes				
vert=DE_VerticalAcceleration	INT	Yes				
yaw=DE_VerticalAcceleration	INT	Yes				
pathHistory=DF_PathHistory	SEQUENCE	No				
initialPosition=DF_FullPositionVector	SEQUENCE	No				
utcTime=DF_DDateTime	SEQUENCE	No				
year=DE_DYear	INTEGER	No				
month=DE_DMonth	INTEGER	No				

PSM Data Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
day=DE_DDay	INTEGER	No				
hour=DE_DHour	INTEGER	No				
minute=DE_DMinute	INTEGER	No				
second=DE_DSecond	INTEGER	No				
offset=DE_DOffset	INTEGER	No				
long=DE_Acceleration	INT	Yes				
lat=DE_Acceleration	INT	Yes				
elevation=DE_Elevation	INT	No				
heading=DE_Heading	INT	No				
speed=DF_TransmissionAndSpeed	SEQUENCE	No				
transmisson=DE_TransmissionState	ENUM	Yes				
speed=DE_Velocity	INT	Yes				
posAccuracy=DF_PositionalAccuracy	SEQUENCE	No				
semiMajor=DE_SemiMajorAxisAccuracy	INT	Yes				
semiMinor=DE_SemiMinorAxisAccuracy	INT	Yes				
orientation=DE_SemiMajorAxisOrientation	INT	Yes				
timeConfidence=DE_TimeConfidence	ENUM	No				
posConfidence=DF_PositionConfidenceSet	SEQUENCE	No				
pos=DE_PositionConfidence	ENUM	Yes				
elevation=DE_ElevationConfidence	ENUM	Yes				
speedConfidence=DF_SpeedHeadingThrottleConfidence	SEQUENCE	No				
heading=DE_HeadingConfidence	ENUM	Yes				
speed=DE_SpeedConfidence	ENUM	Yes				
throttle=DE_ThrottleConfidence	ENUM	Yes				
currGNSSstatus=DE_GNSSstatus	BIT STRING	No				
crumbData=DF_PathHistoryPointList	SEQUENCE	Yes				
latOffset=DE_OffsetLL-B18	INTEGER	Yes				

PSM Data Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
lonOffset=DE_OffsetLL-B18	INTEGER	Yes				
elevationOffset=DE_VertOffset-B12	INTEGER	Yes				
timeOffset=DE_TimeOffset	INTEGER	Yes				
speed=DE_Speed	INTEGER	No				
posAccuracy=DF_PositionalAccuracy	SEQUENCE	No				
semiMajor=DE_SemiMajorAxisAccuracy	INTEGER	Yes				
semiMinor=DE_SemiMinorAxisAccuracy	INTEGER	Yes				
orientation=DE_SemiMajorAxisOrientation	INTEGER	Yes				
heading=DE_CoarseHeading	INTEGER	No				
pathPrediction=DF_PathPrediction	SEQUENCE	No				
PathHistoryPoint=DF_PathHistoryPoint	SEQUENCE	Yes				
latOffset=DE_OffsetLL-B18	INT	Yes				
lonOffset=DE_OffsetLL-B18	INT	Yes				
elevationOffset=DE_VertOffset-B12	INT	Yes				
timeOffset=DE_TimeOffset	INT	Yes				
speed=DE_Speed	INT	No				
posAccuracy=DF_PositionalAccuracy	SEQUENCE	No				
semiMajor=DE_SemiMajorAxisAccuracy	INT	Yes				
semiMinor=DE_SemiMinorAxisAccuracy	INT	Yes				
orientation=DE_SemiMajorAxisOrientation	INT	Yes				
heading=DE_CoarseHeading	INT	No				
radiusOfCurve=DE_RadiusOfCurvature	INT	Yes				
confidence=DE_Confidence	INT	Yes				
propulsion=DF_PropelledInformation	CHOICE	No				
human=DE_HumanPropelledType	ENUM	Yes				
animal=DE_AnimalPropelledType	ENUM	Yes				
motor=DE_MotorizedPropelledType	ENUM	Yes				

PSM Data Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
useState=DE_PersonalDeviceUsageState	BIT STRING	No				
crossRequest=DE_PersonalCrossingRequest	BOOL	No				
crossState=DE_PersonalCrossingInProgress	BOOL	No				
clusterSize=DE_NumberOfParticipantsInCluster	ENUM	No				
clusterRadius=DE_PersonalClusterRadius	INT	No				
eventResponderType=DE_PublicSafetyEventResponderWorkerType	ENUM	No				
activityType=DE_PublicSafetyAndRoadWorkerActivity	BIT STRING	No				
activitySubType=DE_PublicSafetyDirectingTrafficSubType	BIT STRING	No				
assistType=DE_PersonalAssistive	BIT STRING	No				
sizing=DE_UserSizeAndBehaviour	BIT STRING	No				
attachment=DE_Attachment	ENUM	No				
attachmentRadius=DE_AttachmentRadius	INT	No				
animalType=DE_AnimalType	ENUM	No				
regional=regionalExtension	SEQUENCE	No				

2.5 NTCIP 1202 Data Elements

Most traffic signal controllers (TSCs) support the NTCIP 12XX family of standards, such as 1201 and 1202. These standards provide status and control of the TSC through SNMP. The data available from TSCs are based on Management Information Base (MIB) objects. NTCIP data is encoded based on the traffic signal phase, which needs to be translated to an intersection-centric or vehicle-centric encoding. Due to the nature of NTCIP, a single connection to the TSC is recommended. This single management authority is expected to track multiple requests and commands, and it is recommended that the management authority acts as the provider for CVDF messages from the signal controller. NTCIP data elements may be communicated in blocks of data, the structure of the blocks is defined by the underlying standard. While the ICD provides support for NTCIP data blocks, the important components are the data elements themselves. The elements can be provided in blocks or individually, as more data elements are provided the applications will be better supported.

Based on stakeholder engagement, the valuable NTCIP 1202 messages are noted below. These messages or message blocks contain the data elements listed in Table 8, along with their data type and whether the element is required in the NTCIP 1202 standard. The applications that utilize these data elements are also identified, though it is important to note that while a data element may not be directly utilized by an application, the additional data may provide insight about the intersection that contributes to the accuracy of an application. Based on discussions with stakeholders during the concept of operations walkthrough, third-party companies need a way to determine how the timestamps for SPaT messages are synchronized to a disciplined time such as GPS or Network Time Protocol (NTP). Actuated Signal Controllers (ASCs) typically use AC line frequency to establish time stamps, which allows controllers that are connected to the same AC line (such as that along an arterial) to stay synchronized together. This differs from GPS time, which is typically the time source for vehicles, and some way to coordinate the two is needed. As per 1202 v 3 (yet to be ratified), this can be accomplished through the Current Tick Counter and the Current Tick Counter Offset. These two NTCIP objects should be requested as a combined atomic varbind Protocol Data Unit (PDU), which guarantees consistency and provides a subscriber with the information needed to establish the difference between the timestamp on the actuated signal controller (ASC) and a disciplined time source. The ASC SPaT data elements are required in order to provide the high-level information regarding the relative time of the controller that is generating SPaT messages compared to a known synchronized time source such as GPS or NTP stratum-1 synchronized time source. As per NTCIP 1202v0323a, ascCurrentTick defines the local time for the controller in tenths of a second. SPaT data is relative to this local time. The ascCurrentTickMsOffset indicates the elapsed milliseconds from the exact tick when the object was encoded by the ASC. This provides millisecond accuracy for calculating delta times between ASC time and disciplined time for SPaT data.

Table 8: NTCIP 1202 data elements that support applications in the CVDF

NTCIP 1202 Data Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
ASC SPaT						
spatTimestamp	OCTET STRING	Yes	X	X	X	X
ascCurrentTick	INT	Yes	X	X	X	X
ascCurrentTickMsOffset	INT	Yes	X	X	X	X
Phase Parameters						
Phase Status Group Entry						
phaseStatusGroupNumber	INT	No				X
phaseStatusGroupReds	INT	No				X
phaseStatusGroupYellows	INT	No				X
phaseStatusGroupGreens	INT	No				X
phaseStatusGroupDontWalks	INT	No				X
phaseStatusGroupPedClears	INT	No				X
phaseStatusGroupWalks	INT	No				X
phaseStatusGroupVehCalls	INT	No				X
phaseStatusGroupPedCalls	INT	No				X
phaseStatusGroupPhaseOns	INT	No				X
phaseStatusGroupPhaseNexts	INT	No				X
Phase Table Entry						
phaseNumber	INT	No				X
phaseWalk	INT	No				X
phasePedestrianClear	INT	No				X
phaseMinimumGreen	INT	No				X
phasePassage	INT	No				X
phaseMaximum1	INT	No				X
phaseMaximum2	INT	No				X
phaseYellowChange	INT	No				X
phaseRedClear	INT	No				X
phaseRedRevert	INT	No				X
phaseAddedInitial	INT	No				X
phaseMaximumInitial	INT	No				X
phaseTimeBeforeReduction	INT	No				X
phaseCarsBeforeReduction	INT	No				X
phaseTimeToReduce	INT	No				X
phaseReduceBy	INT	No				X
phaseMinimumGap	INT	No				X
phaseDynamicMaxLimit	INT	No				X

NTCIP 1202 Data Elements	Type	Req	Applications RLVW	PedX	ECO	X-ECO
phaseDynamicMaxStep	INT	No				X
phaseStartup	INT	No				X
phaseOptions	INT	No				X
phaseRing	INT	No				X
phaseConcurrency	OCTET STRING	No				X
Coordination Parameters						
coordMaximumMode	INT	No				X
coordForceMode	INT	No				X
coordPatternStatus	INT	No				X
localFreeStatus	INT	No				X
coordCycleStatus	INT	No				X
coordSyncStatus	INT	No				X
Pattern Table Entry						
patternNumber	INT	No				X
patternCycleTime	INT	No				X
patternOffsetTime	INT	No				X
patternSplitNumber	INT	No				X
patternSequenceNumber	INT	No				X
Split Table Entry						
splitNumber	INT	No				X
splitPhase	INT	No				X
splitTime	INT	No				X
splitMode	INT	No				X
splitCoordPhase	INT	No				X
Overlap Parameters						
Overlap Entry						
overlapNumber	INT	No				X
overlapType	INT	No				X
overlapIncludedPhases	OCTET STRING	No				X
overlapModifierPhases	OCTET STRING	No				X
overlapTrailGreen	INT	No				X
overlapTrailYellow	INT	No				X
overlapTrailRed	INT	No				X
Overlap Status Group Entry						
overlapStatusGroupNumber	INT	No				X
overlapStatusGroupReds	INT	No				X
overlapStatusGroupYellows	INT	No				X

NTCIP 1202 Data Elements	Type	Req	Applications			
			RLVW	PedX	ECO	X-ECO
overlapStatusGroupGreens	INT	No				X
Ring Parameters						
Ring Status Table Entry						
ringStatus	INT	No				
Unit Parameters						
Special Function Output Entry						
specialFunctionOutputNumber	INT	No				
Channel Parameters						
Channel Table Entry						
maxChannels	INT	No	X	X	X	X
channelNumber	INT	No	X	X	X	X
channelControlSource	INT	No	X	X	X	X
channelControlType	INT	No	X	X	X	X
channelFlash	INT	No	X	X	X	X
channelDim	INT	No	X	X	X	X
channelStatusGroupNumber	INT	No	X	X	X	X
channelStatusGroupReds	INT	No	X	X	X	X
channelStatusGroupYellows	INT	No	X	X	X	X
channelStatusGroupGreens	INT	No	X	X	X	X

2.6 TMDD 3.03d

The Traffic Management Data Dictionary (TMDD) defines data elements that correspond to the data that is typically transmitted between traffic management centers (TMCs). Due to this, the TMDD provides data descriptions spanning many ITS devices, as listed within their "Object Classes." Much of this data is not intended for low latency data communication since Center-to-Center use cases are typically high latency connections. The TMDD is very expansive, with more than one thousand pages of system management data documentation. A subset of these data elements needs to be supported by a CVDF. The other object classes found within TMDD can likely be added upon establishing need, subject to availability of these datasets within the ATMS system. Based on stakeholder engagement, the data elements in Table 9 are the data elements and data types from the TMDD that are directly relevant to a CVDF implementation.

Table 9: TMDD data elements

TMDD v3.03d Data Elements		Type	Req
Intersection			
	actuation-mode	ENUM	Yes
	offset-reference	ENUM	Yes
	planned-signal-timing-mode	ENUM	Yes
	signal-control-mode	ENUM	Yes
	signal-control-source	ENUM	Yes
	signal-timing-mode	ENUM	Yes
	turning-movement-code	ENUM	Yes
	timing-duration	INT	Yes
Time			
	reference-code	ENUM	Yes

3. CVDF OpenAPI

Web based data portals allow clients to manage various data types, typically called resources, by exposing an API such as Representational State Transfer (REST)[1] or GraphQL[2]. A CVDF is required to support at least one of these architectures (REQ-EXT-1.3). Most of the web services today expose a RESTful API based on HTTP protocol, and typically enforce HTTPS connections to secure traffic.[3] In addition to encrypting traffic, authentication mechanisms such as JSON Web Tokens (JWT) are utilized to provide a means of authentication and access control to resources.

SwRI has created a RESTful API that supports the SAE J2735 data elements as an example implementation. The API documentation adheres to the OpenAPI Specification (OAS) and will assist with the development and deployment of consumers and providers that are communicating CV data through a CVDF. Document generation tools, such as Swagger™, can be used to generate clients using the API specification[4]. This capability significantly increases the rate of consumer adoption as well as easing transitions from version to version. REST endpoints are available for message frames and other supported data types. Supported operations include resource creation, modification and deletion via POST, PUT and DELETE requests, respectively, by authorized providers. Resources can be retrieved via GET methods by authorized consumers. The API is versioned to support backwards-compatibility and allow consumers and providers to update at their convenience. The example reference API handles all of the SAE J2735 data

[1] https://restfulapi.net/
[2] https://graphql.org/
[3] https://www.mulesoft.com/resources/api/restful-api
[4] https://swagger.io/

elements that are required, data elements from NTCIP 1202 and TMDD would be implemented in future revisions.

Through this work, we have established the documentation that developers can use as a basis for a client-side or server-side implementation. This method facilitates ease-of-development, as client Software Development Kits (SDKs) in a variety of languages can be directly generated from this document see (Figure 6), as well as server stubs (see Figure 7).

The screenshots below (see Figure 8, Figure 9 , Figure 10, Figure 11 , Figure 12, and Figure 13) provide a visual summary of the API for SAE J2735 data elements. Implementers can interact directly with the API at the following web address:

https://app.swaggerhub.com/apis-docs/SwRI/3rd_party_API_ICD/v1.0.1

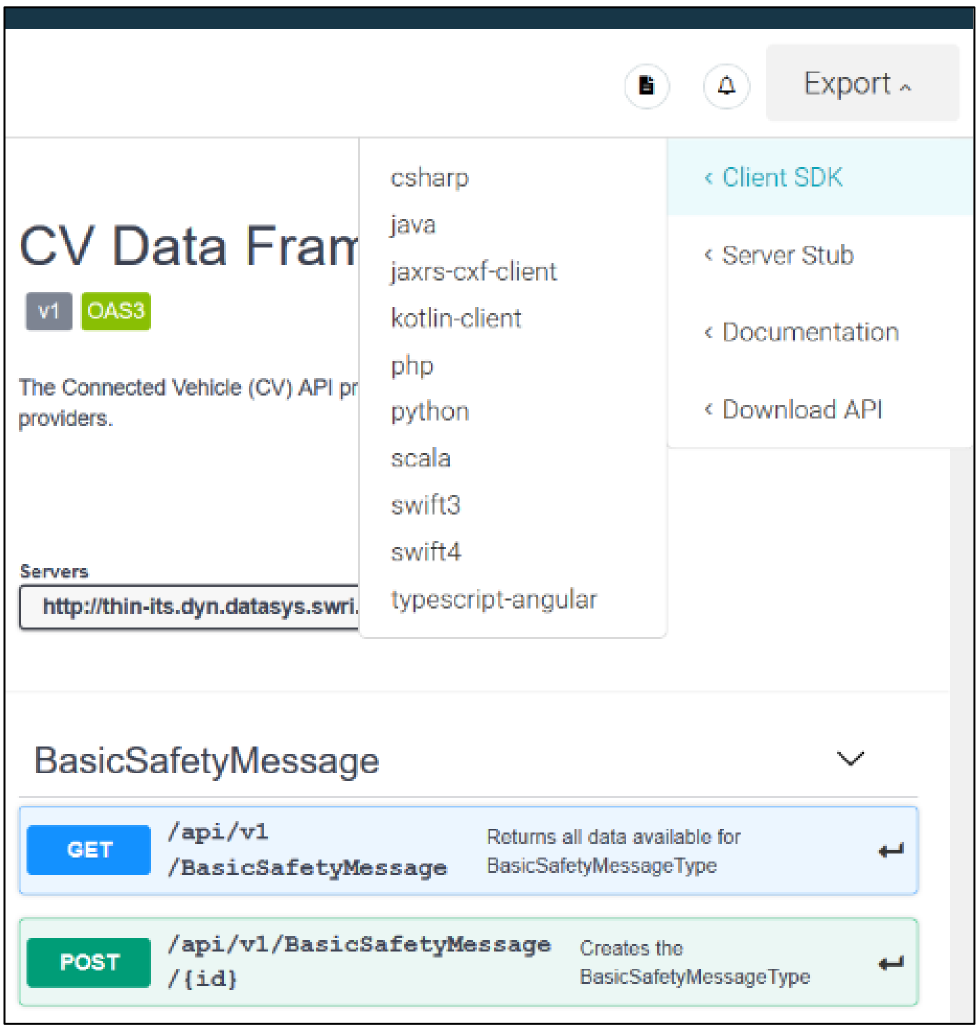

Figure 6: Available options for downloading a client SDK based on the CVDF 1.0.0 API

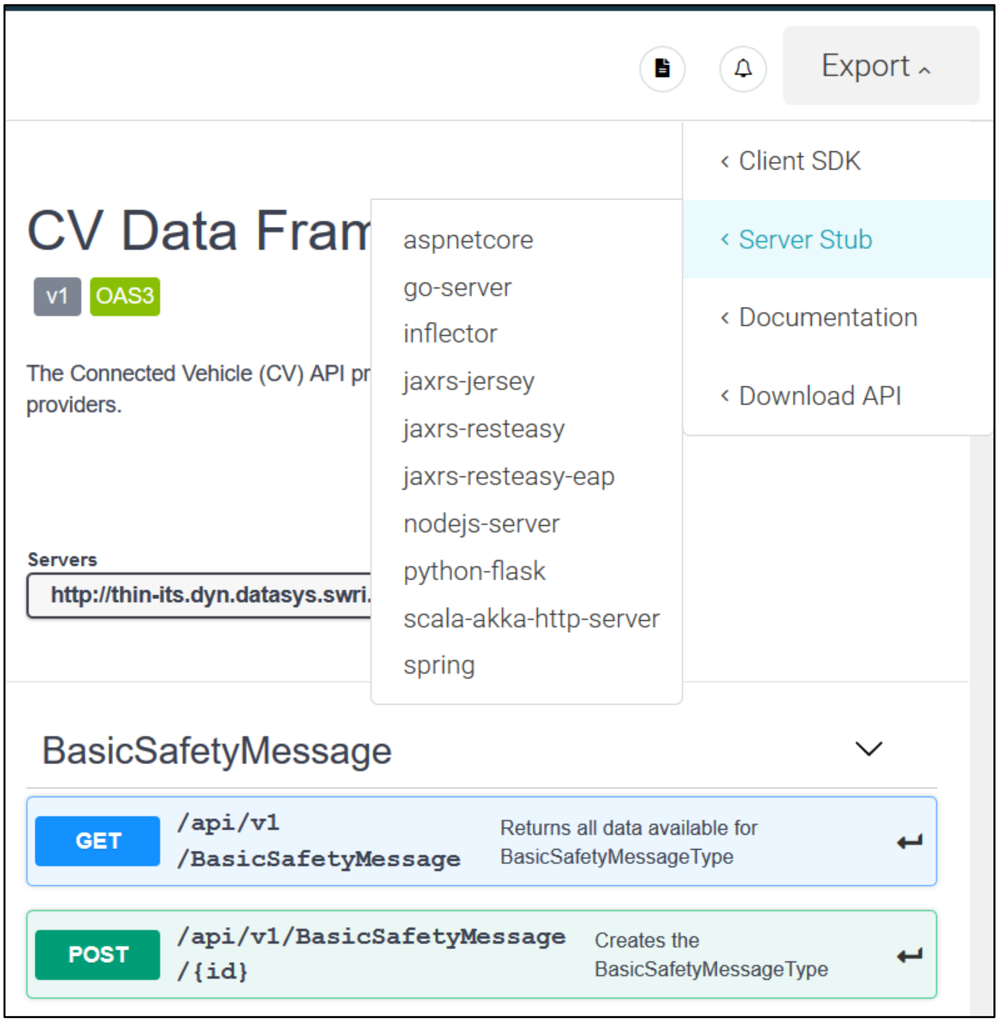

Figure 7: Available options for downloading the server stubs for CVDF 1.0.0

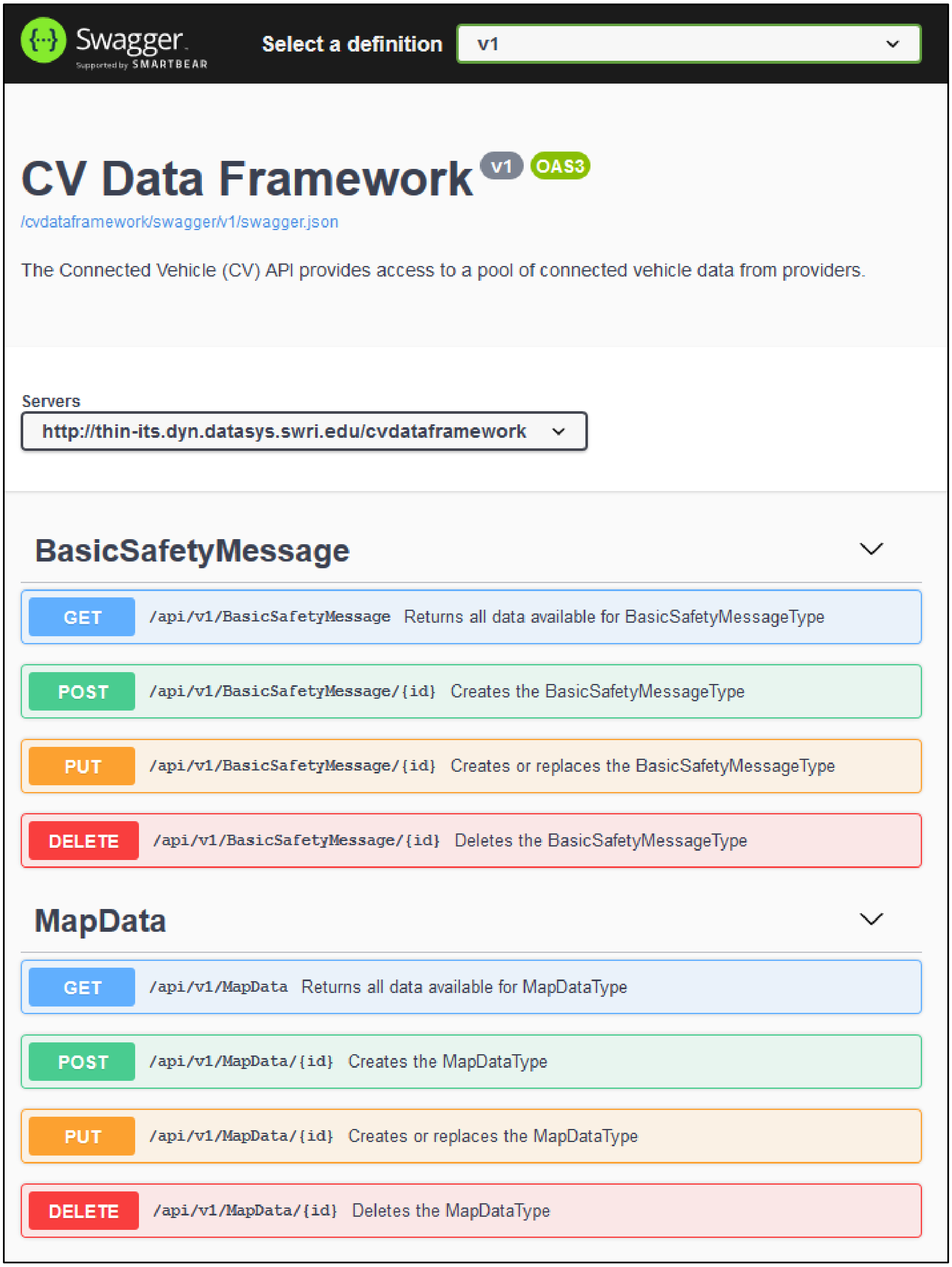

Figure 8: REST API for BasicSafetyMessage and MapData

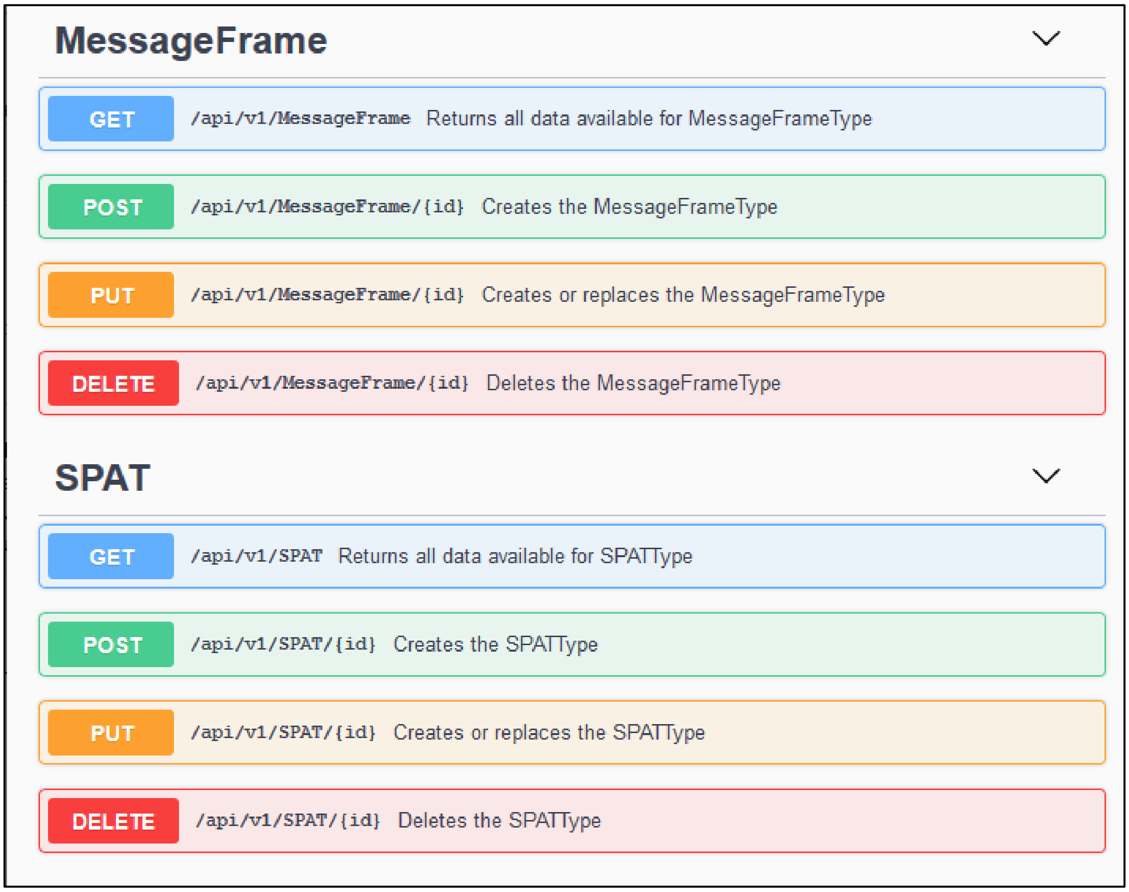

Figure 9 : REST API for MessageFrame and SPAT

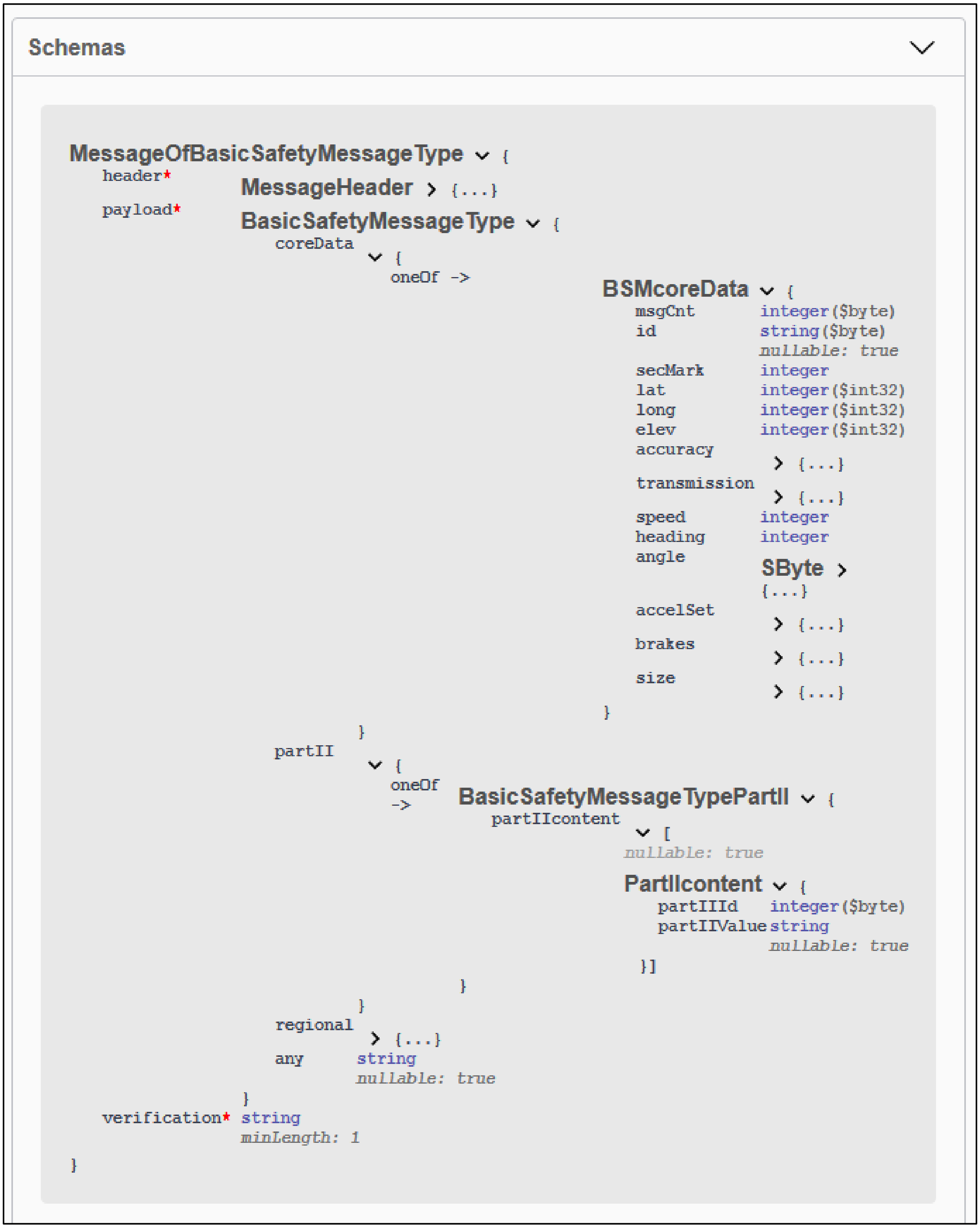

Figure 10: Schema for a CVDF message with a BSM as the payload

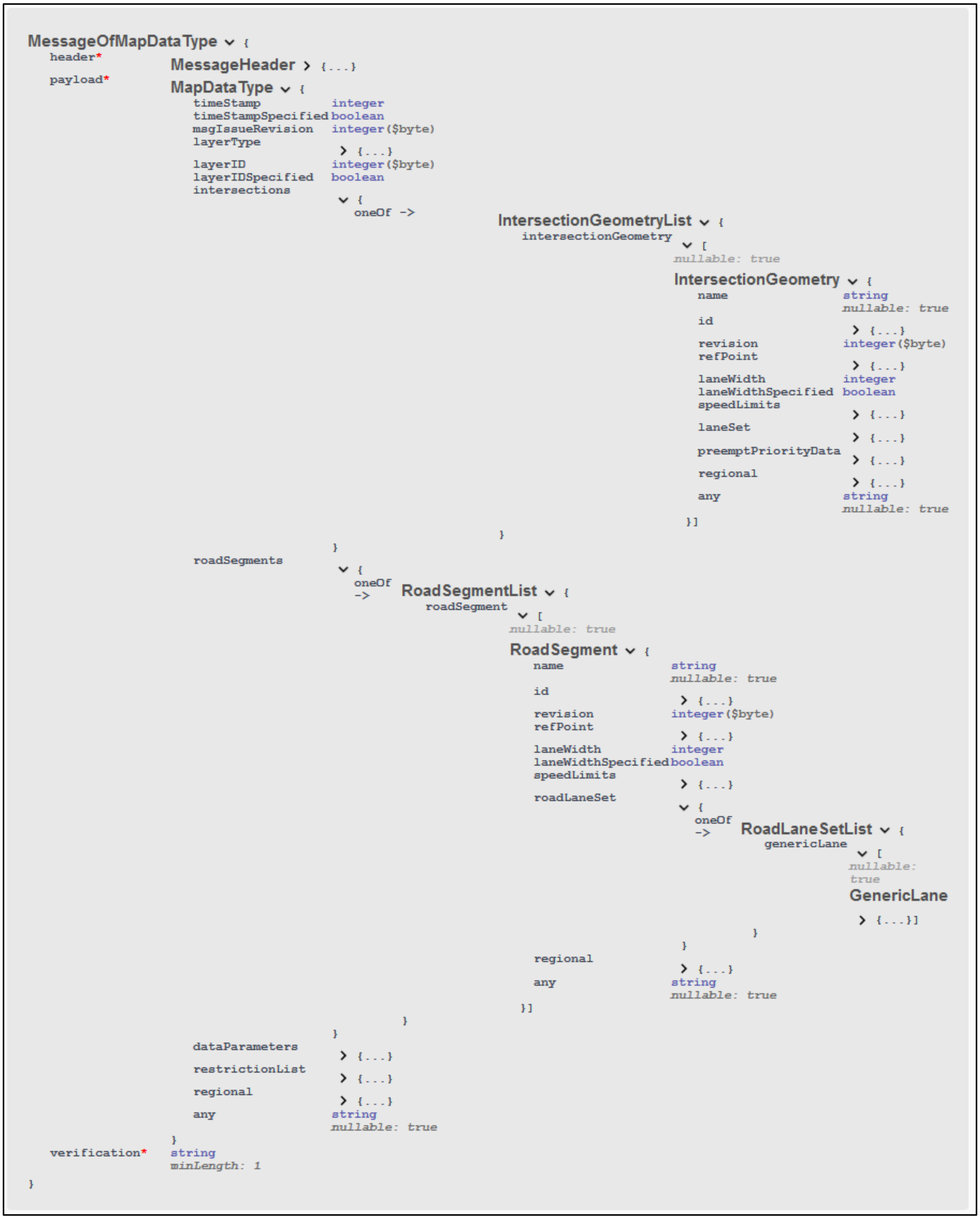

Figure 11 : Schema for a CVDF message with a MAP as the payload

```
MessageOfMessageFrameType ⌄ {
   header*            MessageHeader > {...}
   payload*           MessageFrameType ⌄ {
                         messageId          integer
                         value              string
                                            nullable: true
                         any                string
                                            nullable: true
                      }
   verification*      string
                      minLength: 1
}
```

Figure 12: Schema for a CVDF message with a MessageFrame as the payload

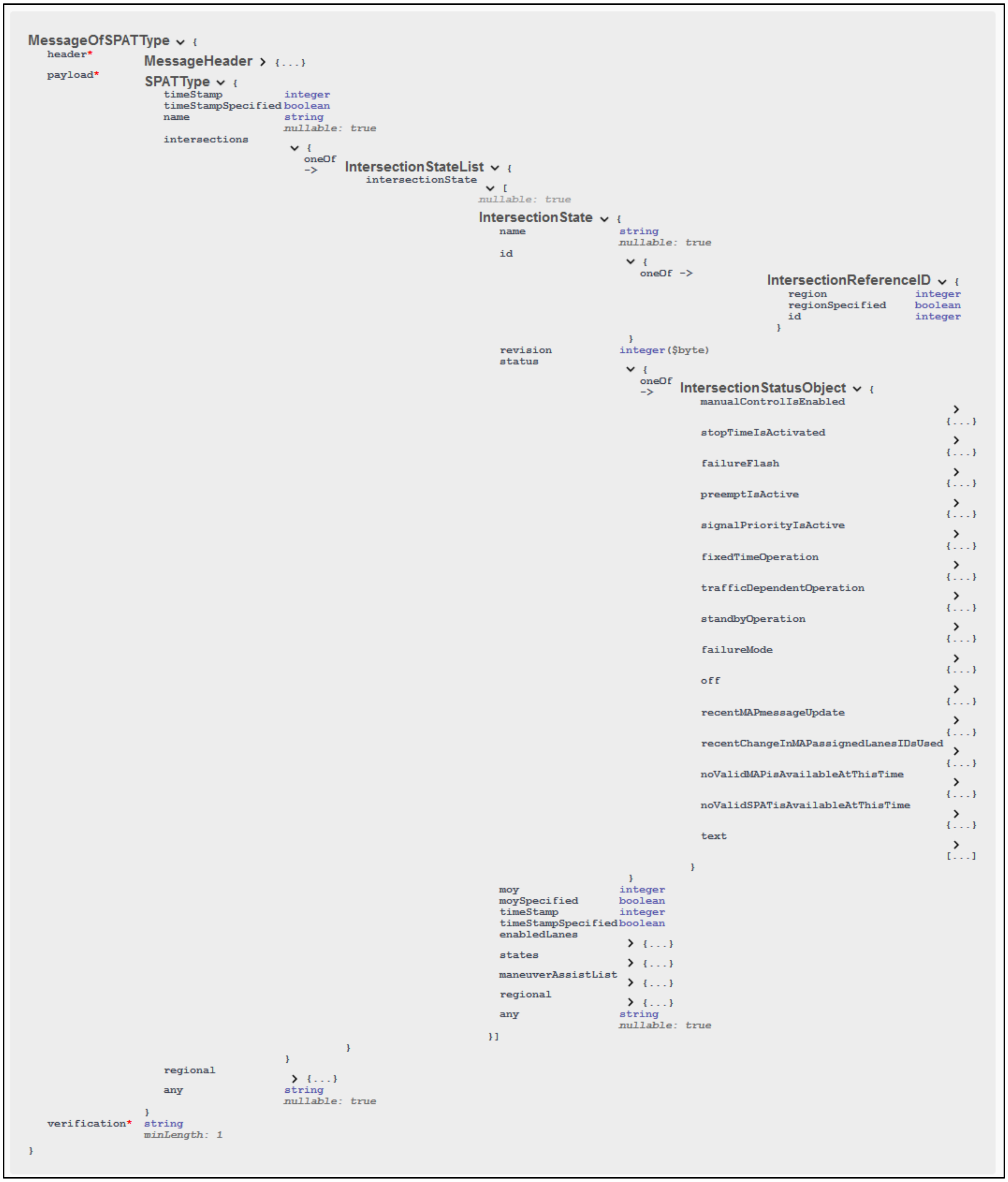

Figure 13: Schema for a CVDF message with a SPaT as the payload

4. Conclusion

The data elements from J2735, NTCIP 1202, and TMDD 3.03d have been identified along with the enabled applications. The data elements that were selected are based on the project objectives as well as stakeholder feedback. An example API in OpenAPI 3.0.0 that supports the J2735 data elements has been provided that allows states and IOOs to communicate CV data to third parties and OEMs. Feedback from stakeholders has been incorporated via email and a virtual walk-through. Through this ICD, data elements that are used by a CVDF implementation and the API to communicate them have been established.